PROJECT#1
EVALUATING HOW CACHE DESIGN CHOICES AFFECT THE PERFORMANCE OF X86 ARCHITECTURE BASED MICROPROCESSOR USING GEM5 SIMULATOR

UNDER: PROF DR. KANAD BASU

TEAM: (100/100)
ABHINAY DWADASI (AXD220054)

YAGNA SRINIVASA HARSHA ANNADATA (YXA210024)

(CE6304) CLASS OF 2023-24 (SPRING SEMESTER)

PROJECT DESCRIPTION:

- Fine tuning the cache hierarchy of X86 architecture using GEM5 simulator.
- The following Design parameters have been manipulated to evaluate the performance of the CPU as specified:
- **CPU type**: Timing
- **Cache levels**: Two levels; L1 and L2 caches.
- **Cache sizes considered:**
- 128 kB and 256 kB for L1 data cache.
- 64 kB and 256 kB for L1 Instruction cache.
- 512 and 1024 kB for L2 cache.
- **Associativity:**
- For L1D and L2I: direct mapped and 4-way Set associativity
- For L2: Direct mapped and 8-way set associativity.
- Cache Block Size: 32 and 64 bytes

GEM5 CONFIGURATION PART-1

- Gem5 was accessed from the virtual server CE6304.utdallas.edu
- Instead of accessing the Gem5 directly on the server we have copied the files from Gem5 directory (/usr/local/gem5). Copying the configuration files and moving them to our directory was done using “cp rf-” command.
- We then built the X86 processor in the copied directory on our server using no machine. The command used was (scons build/X86/gem5.opt)
- We then downloaded all the 5 bench mark files from Github using the link: https://github.com/timberjack/Project1_SPEC
- We then saved all the files in a folder called bench marks in the same Gem5 directory.

GEM5 CONFIGURATION PART-2

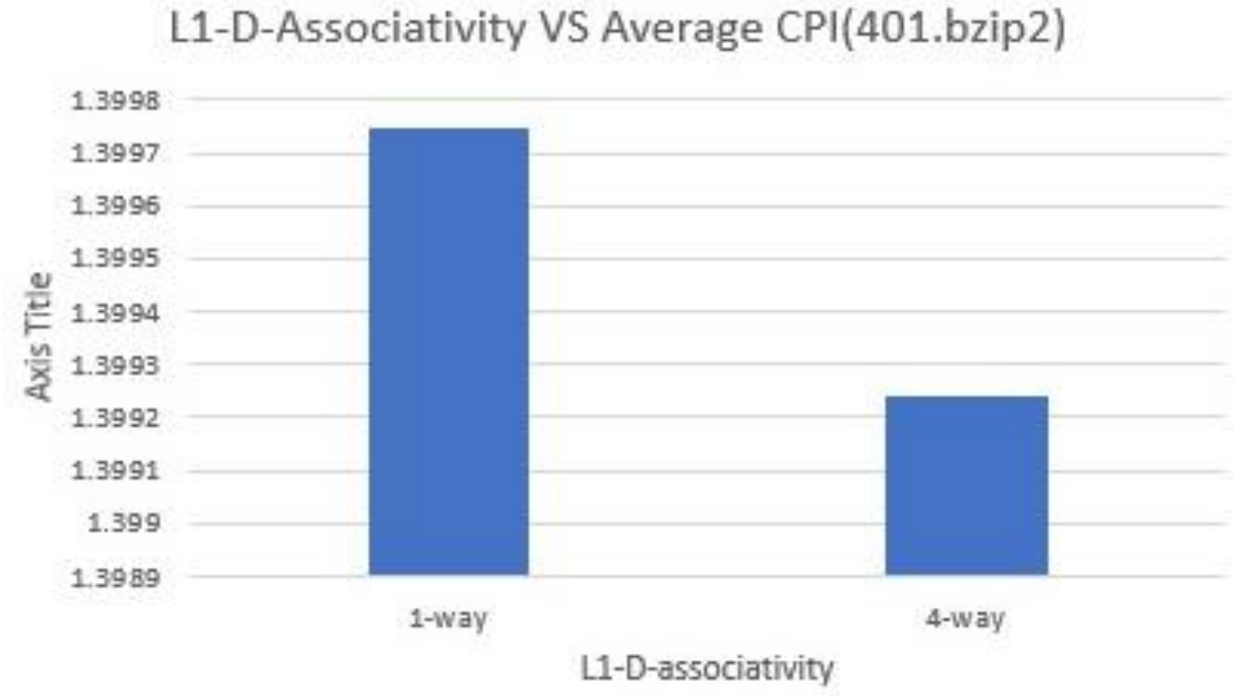
- In order to simulate X86 based CPU we, use the following benchmarks specified: 401.bzip2, 429.mcf, 456.hmmer, 458.sjeng and 470.lbm
- An Output directory is also created to save all the Stats.txt files which are required to calculate the CPI
- We will be using System Call Emulation mode as prescribed using the script in the path: "/configs/example/se.py".
- The Number of instructions chosen is 300000000 for all benchmarks except 429.mcf which has 75000000.
- All the input variables are dynamically allocated according to the chosen configuration.

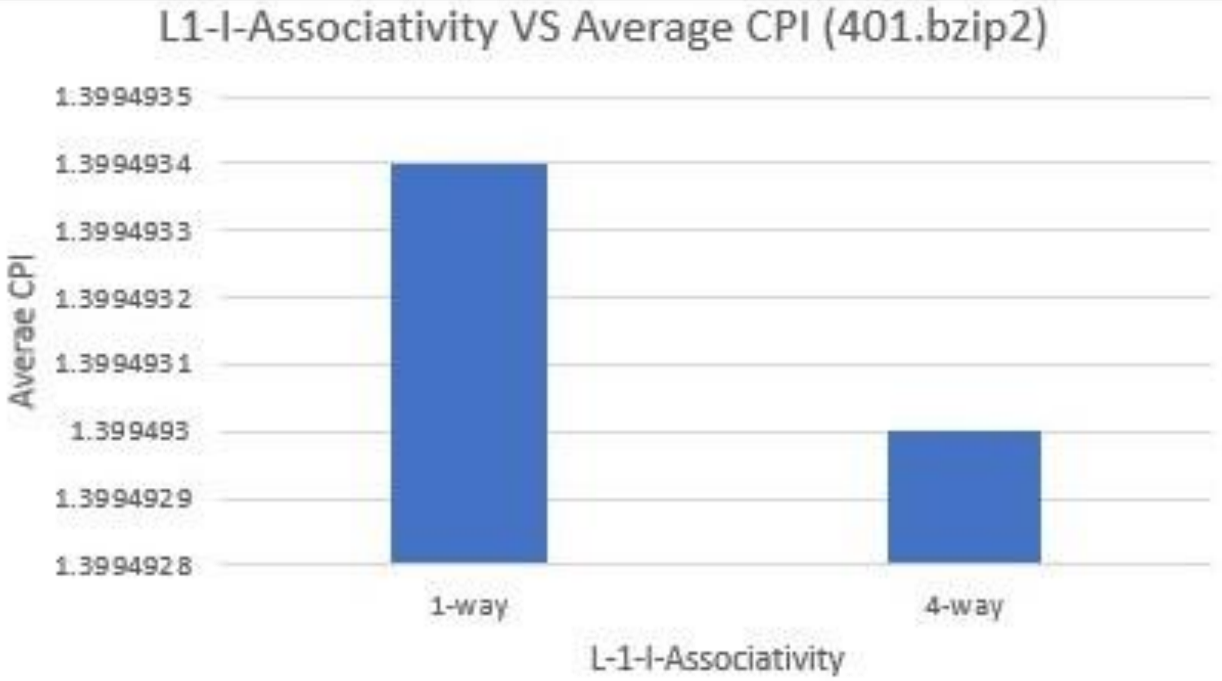
GEM-5 CONFIG3URATION PART-3

- Outputs are stored in the local machine with respective names and configuration in their title.
- We have written a python code to generate the output of respective benchmark which is also attached in the software package.
- Also, we have written python code to calculate the CPI of the generated output files which saves the data in two text file in sorted and unsorted formats.

OPTIMUM CPI FOR 401.BZIP2

- The minimum CPI obtained was : 1.37867333
- The memory configuration is as follows:
  - Block size : 64
  - L1 Data Cache Associativity: 4
  - L1 Instruction Cache Associativity: 4
  - L2 Cache Associativity: 8
  - L1 Data Cache Size: 256kB
  - L1 Instruction Cache Size: 256kB
  - L2 Cache Size: 1mB



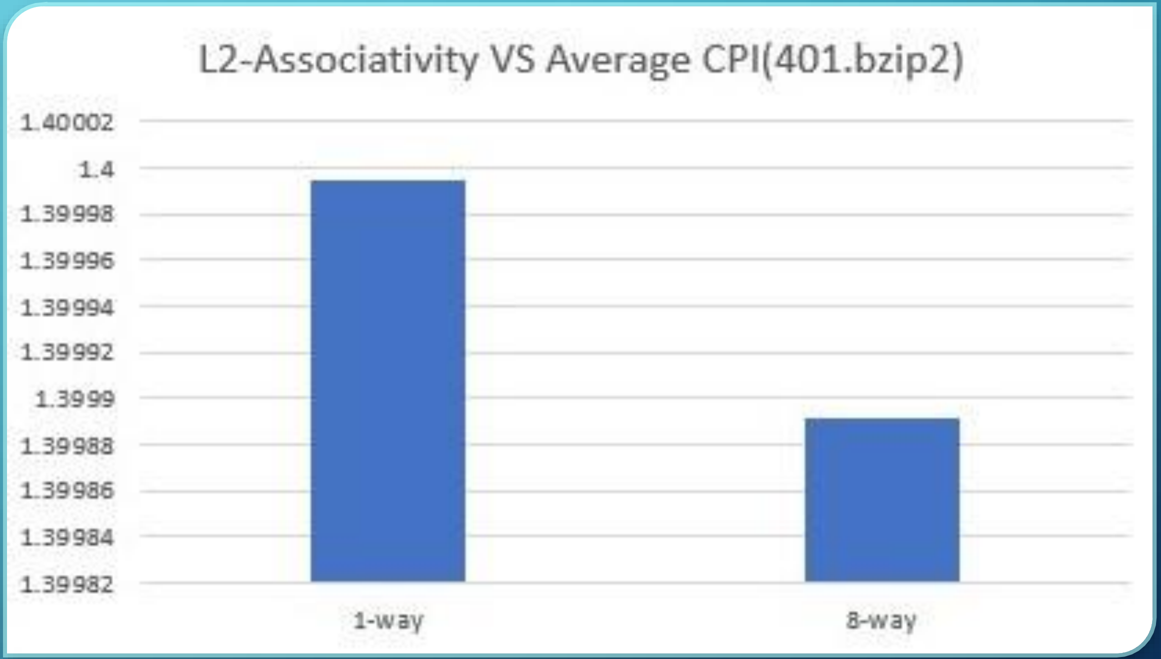


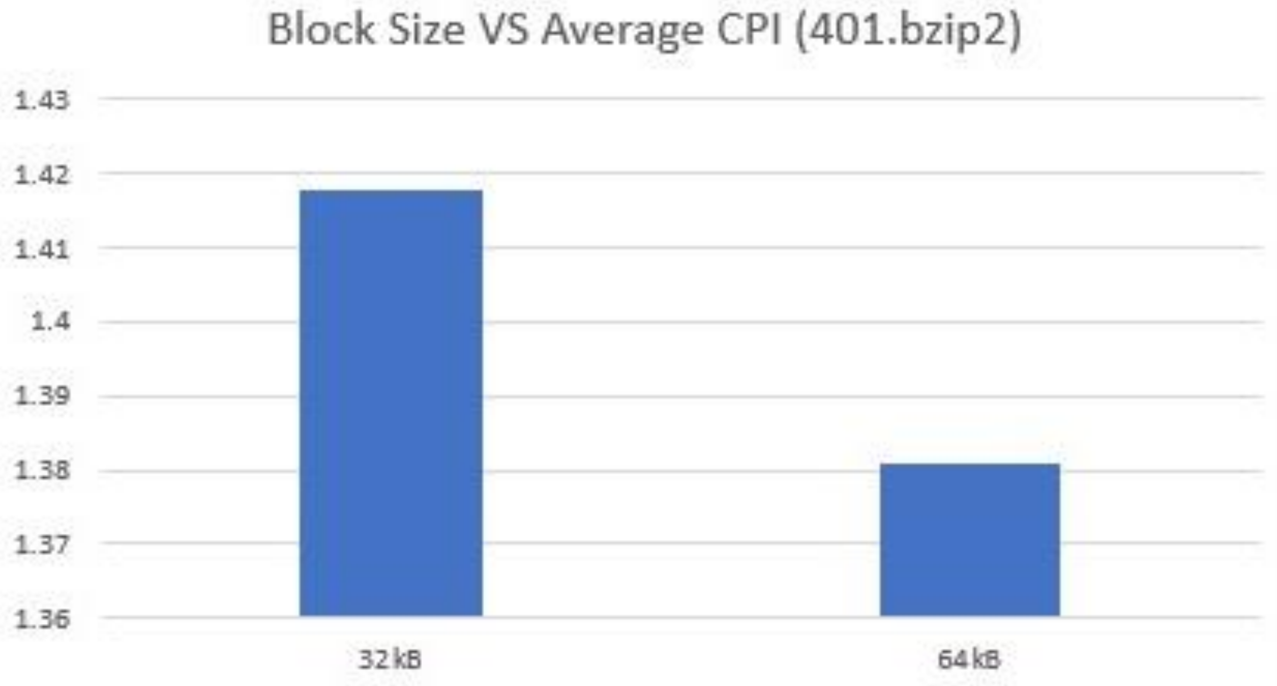
GRAPHICAL ANALYSIS OF L1 ASSOCIATIVITY FOR 401.BZIP2

- As seen in graphs, from the derived values using 401.bzip2 benchmark and the average CPI calculated, following conclusions can be made:
- The performance of a CPU with direct mapped L1-I and L-1D cache shows considerably lower performance when compared with a 4-way set associative L1-I and L1-d cache.
- This mean that, the more the associativity the lower the CPI and the better the performance.

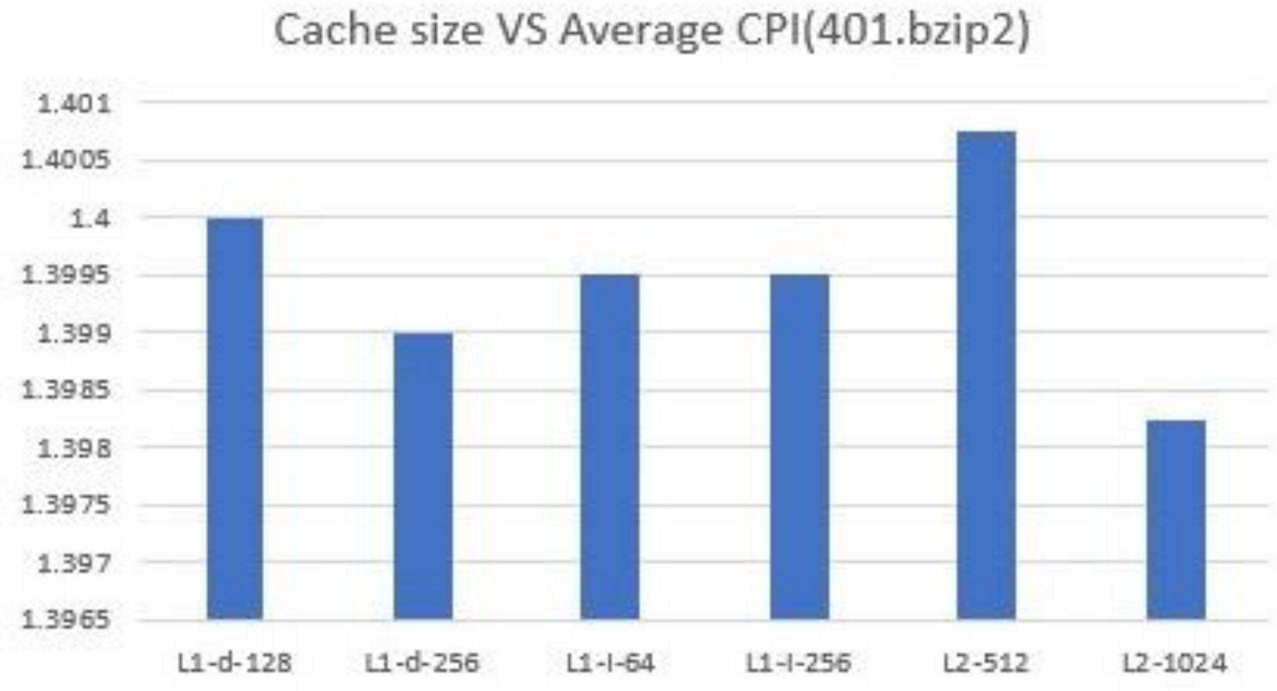
GRAPHICAL ANALYSIS OF L2 ASSOCIATIVITY FOR 401.BZIP2

- From the simulation results obtained by running the 401.bzip2 benchmark the following observations about the L2 associativity can be made:
- The direct mapped L2 cache displays considerably less performance.
- The 8-way set associativity of the L2 cache has better performance and less number of misses.
- The more the associativity of L2 cache the better the performance.





Block Size VS Average CPI (401.bzip2)



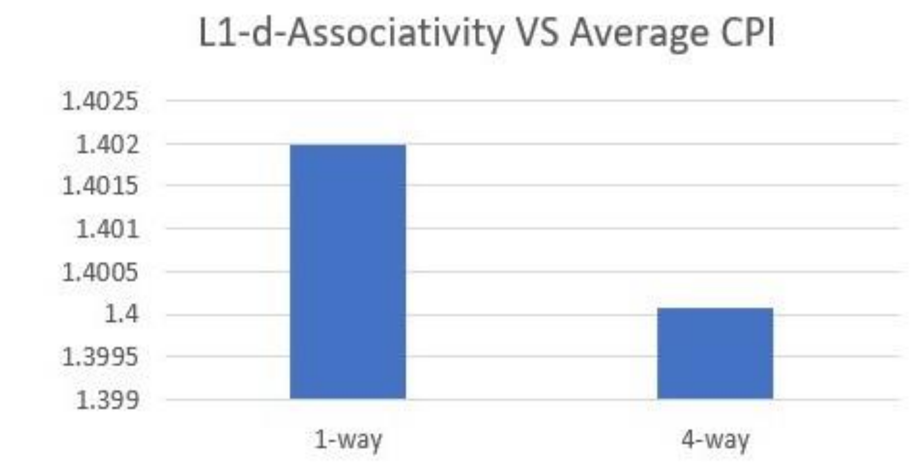
Cache size VS Average CPI(401.bzip2)

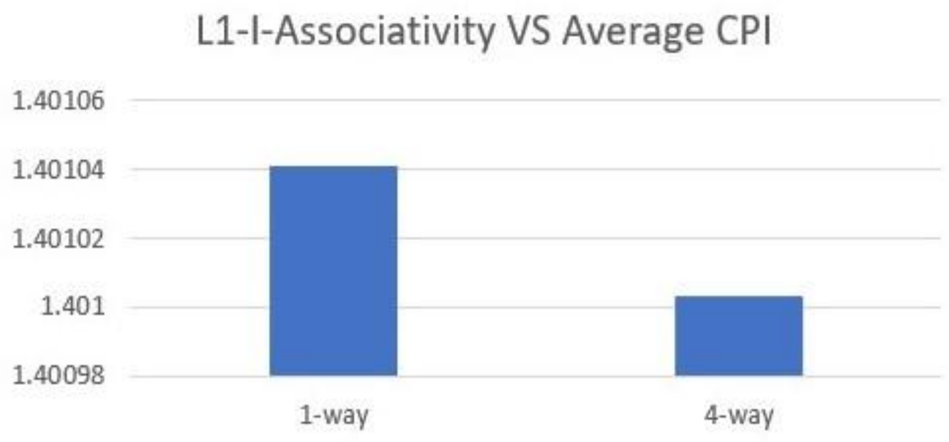
GRAPHICAL ANALYSIS ON CACHE AND BLOCK SIZE FOR 401.BZIP2

- With 401.bzip2 benchmark it is observed the block size has considerably more affect on the performance the than the cache size.
- **Conclusion:** Hence when it comes to size of multiple levels of Cache and the block, the performance is directly proportional to the Size.

OPTIMUM CPI FOR 429.MCF

- The minimum CPI obtained was : 1.3821154086666667
- There are two configurations which had the similar CPI the only difference was the L1 Instruction cache size.
- The memory configuration is as follows:
  - Block size : 64
  - L1 Data Cache Associativity: 4
  - L1 Instruction Cache Associativity: 4
  - L2 Cache Associativity: 8
  - L1 Data Cache Size: 256kB
  - L1 Instruction Cache Size: 256kB,64kB
  - L2 Cache Size: 1mB

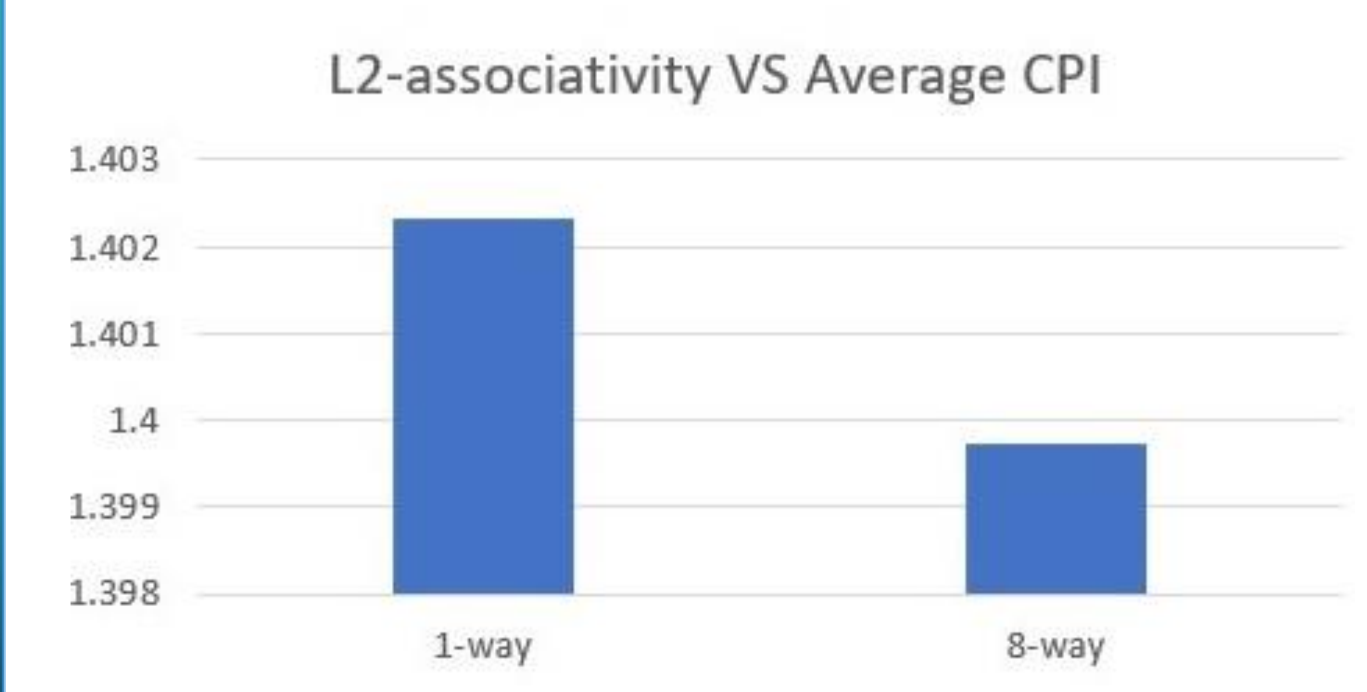




GRAPHICAL ANALYSIS OF L1 ASSOCIATIVITY FOR 429.MCF

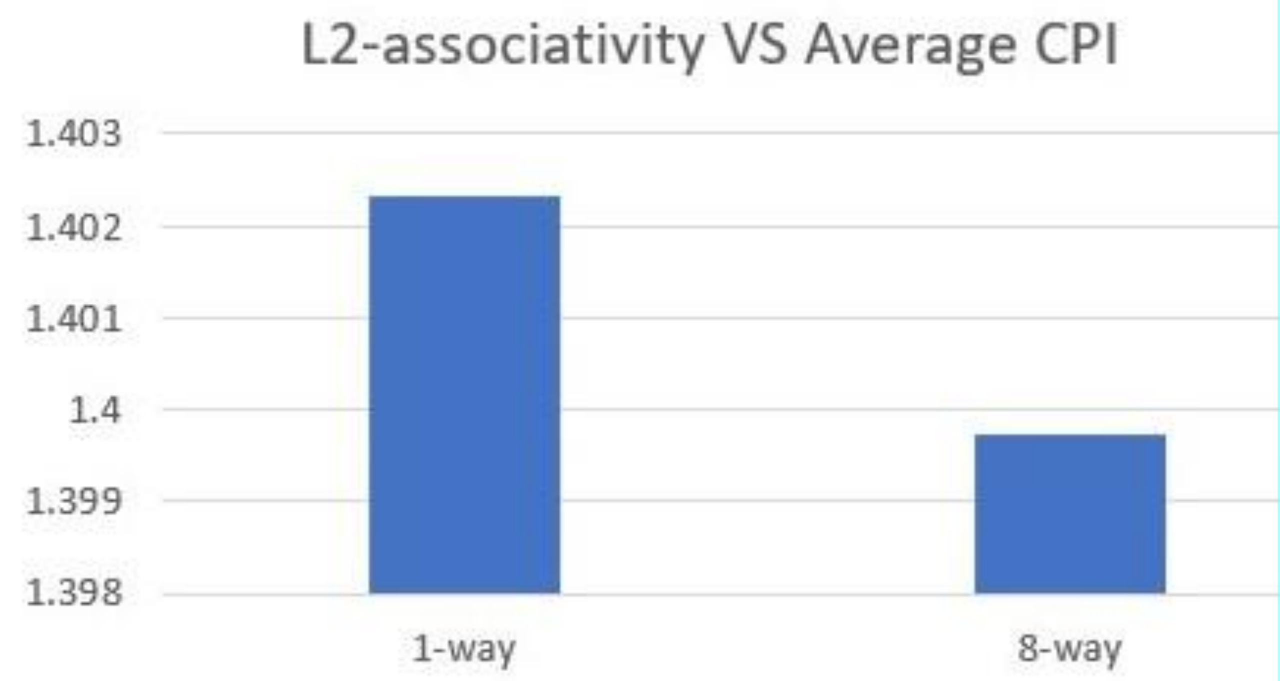
- As seen in graphs, from the derived values using 401.bzip2 benchmark and the average CPI calculated, following conclusions can be made regarding the associativity:
- For both L1I and L1D caches the performance of the processor is directly affected by the associativity.
- **Conclusion:** As the associativity increases the performance of the processor also increase which means the CPI Decreases as we increase the associativity of caches at multiple levels..

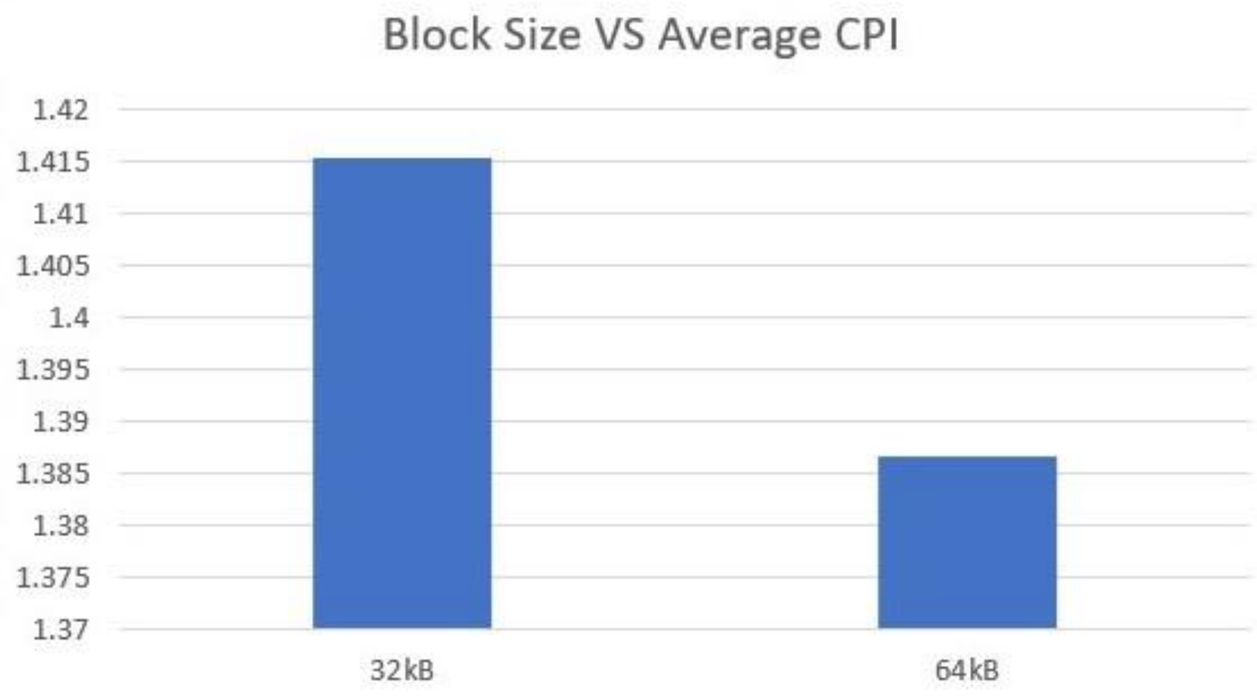
GRAPHICAL ANALYSIS OF L2 ASSOCIATIVITY FOR 429.MCF



- The simulation values obtained by running the 429.mcf benchmark the following observations about the L2 associativity can be made:
- The direct mapped L2 cache displays considerably less performance.
- The 8-way set associativity of the L2 cache has better performance and less number of misses.
- **Conclusion:** The more the associativity of L2 cache the better the performance.

GRAPHICAL ANALYSIS FOR 429.MCF FOR CACHE AND BLOCK SIZE

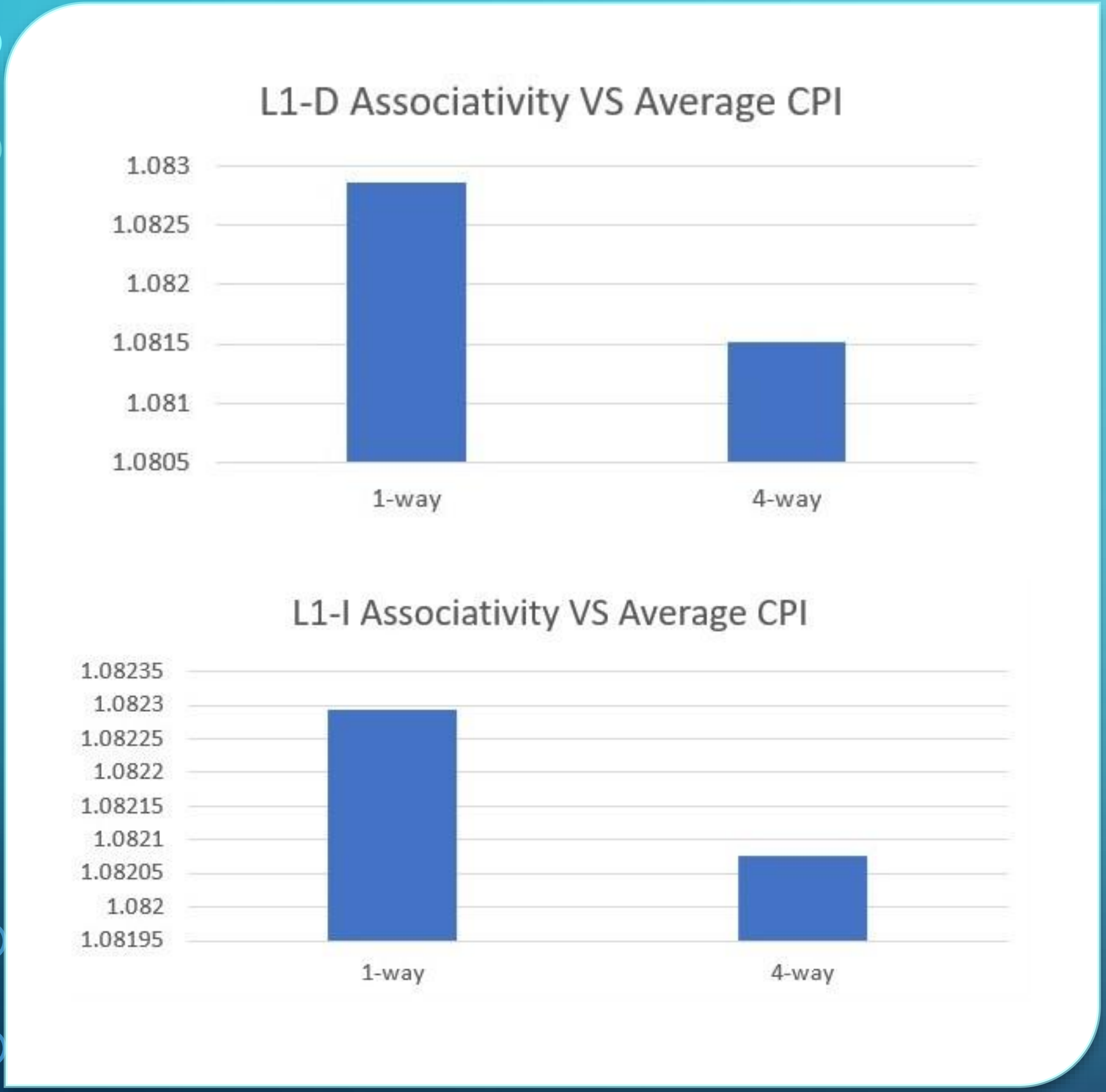




- Using 429.mcf benchmark it is observed the block size has considerably more affect on the performance the than the cache size.
- **Conclusion:** When it comes to size of multiple levels of Cache and the block, the performance is directly proportional to the Size.

OPTIMUM CPI FOR 456.HMMER

- The minimum CPI obtained was : 1.0180928533333333
- The memory configuration is as follows:
  - Block size : 64
  - L1 Data Cache Associativity: 4
  - L1 Instruction Cache Associativity: 4
  - L2 Cache Associativity: 8
  - L1 Data Cache Size: 256kB
  - L1 Instruction Cache Size: 256kB
  - L2 Cache Size:1mB
- There are three configurations which had the similar CPI the other 2 are, L2 cache size at 512kB and rest of the configurations being the same.
- L2 cache associativity at 1 with having all the other configurations same.

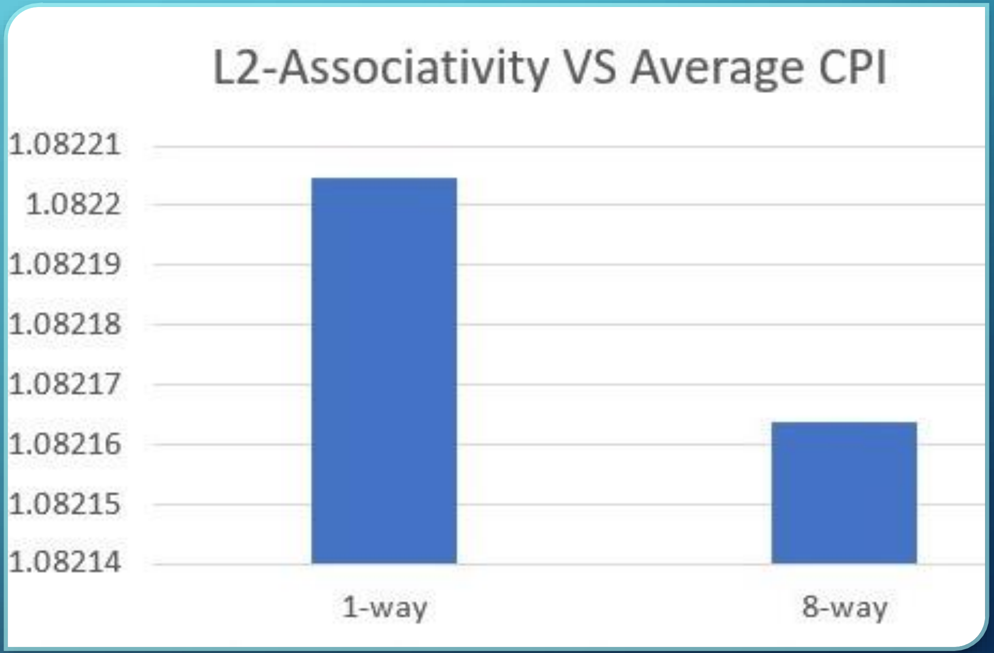


GRAPHICAL ANALYSIS OF L1 ASSOCIATIVITY FOR 456.HMMER

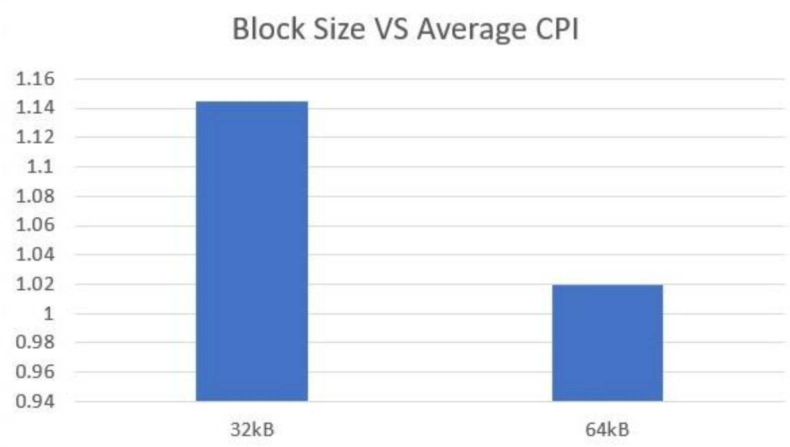
- By running 456.hmmer benchmark and the average CPI calculated, following conclusions can be made regarding the associativity:
- Both L1I and L1D caches the performance of the processor is directly affected by the associativity.
- **Conclusion:** As the associativity increases the performance of the processor also increase which means the CPI decreases

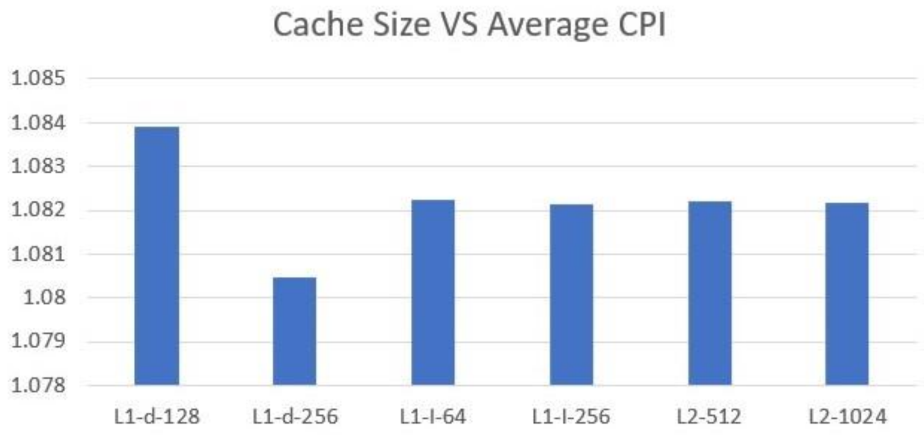
GRAPHICAL ANALYSIS OF L2 ASSOCIATIVITY FOR 456.HMMER

- By running the 456.hmmer benchmark the following observations about the L2 associativity were made:
- It can be observed that direct mapped L2 cache reflects considerably less performance.
- 8-way set associativity of the L2 cache results in better performance and decreased number of misses.
- **Conclusion:** Associativity of L2 cache is proportional to the performance.



GRAPHICAL ANALYSIS OF CACHE SIZE AND BLOCK SIZE FOR 456.HMMER

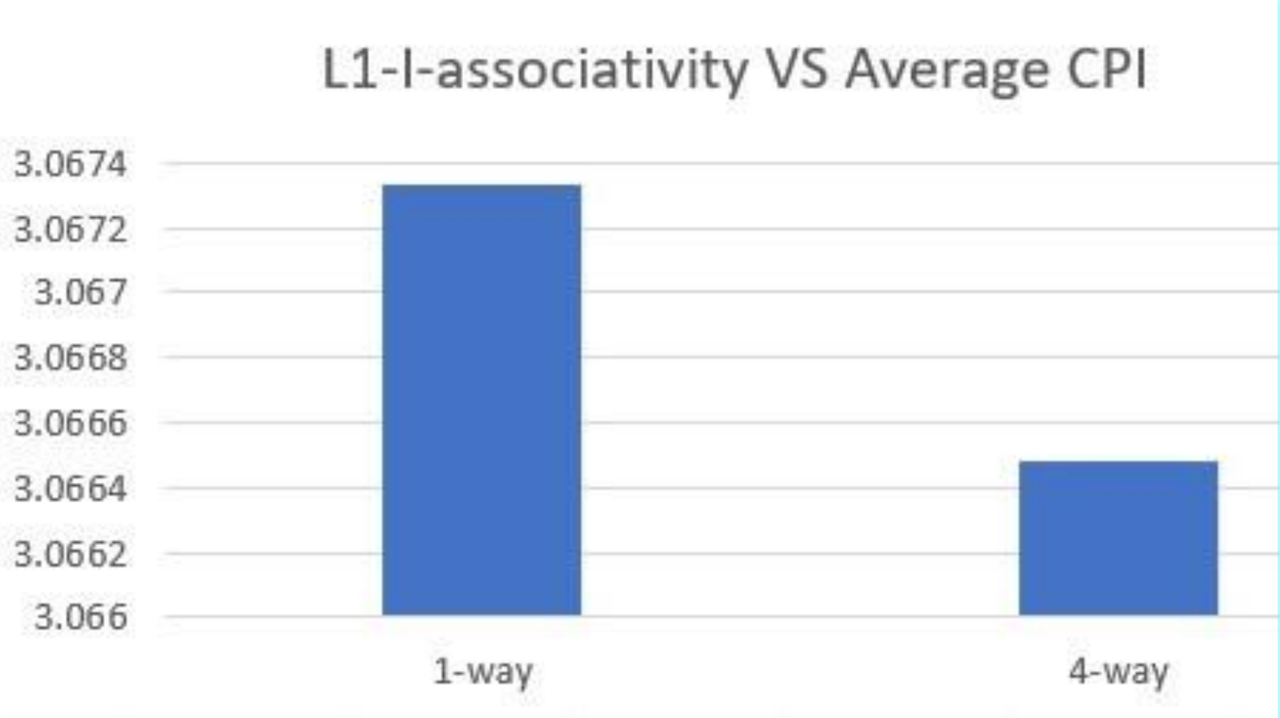

Block Size VS Average CPI

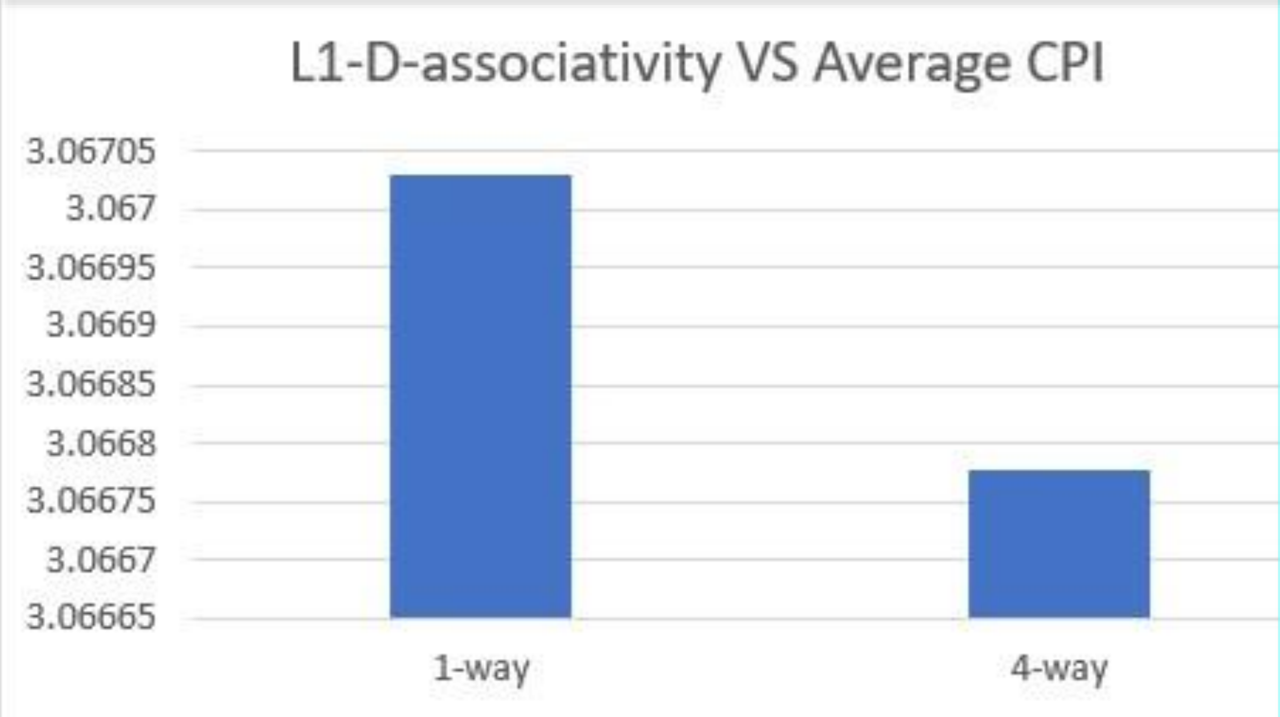

Cache Size VS Average CPI

- 456.hmmer benchmark when run in system emulation mode, it is deduced that the block size has considerably more affect on the performance the than the cache size. We can see that CPI decreases by 8% when block size is doubled
- **Conclusion:** When it comes to size of multiple levels of Cache and the block, the performance is directly proportional to the Size of the Cache as well as the logical blocks.

OPTIMUM CPI FOR 458.SJENG

- The minimum CPI obtained was : 1.48886607
- The memory configuration is as follows:
  - Block size : 64
  - L1 Data Cache Associativity: 4
  - L1 Instruction Cache Associativity: 4
  - L2 Cache Associativity: 8
  - L1 Data Cache Size: 256kB
  - L1 Instruction Cache Size: 256kB
  - L2 Cache Size:1mB

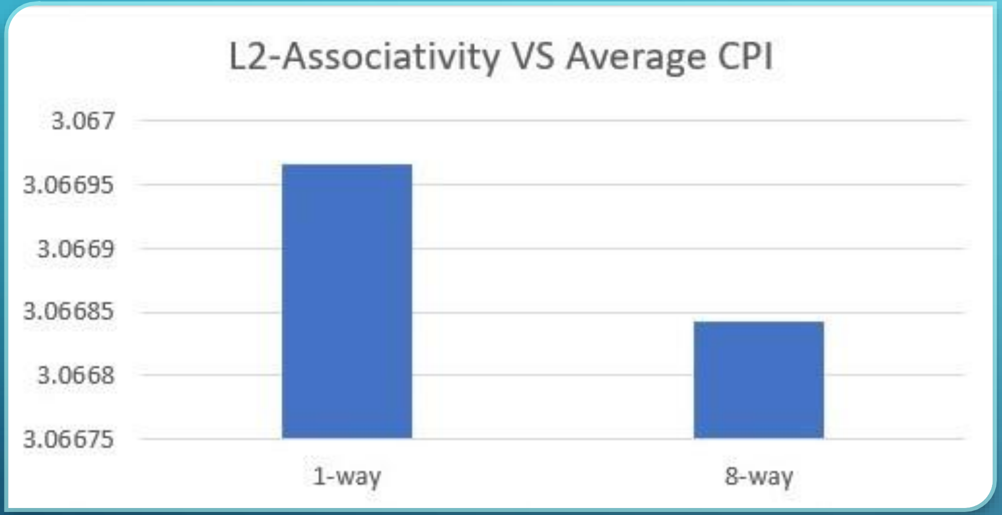




GRAPHICAL ANALYSIS OF L1 ASSOCIATIVITY FOR 458.SJENG

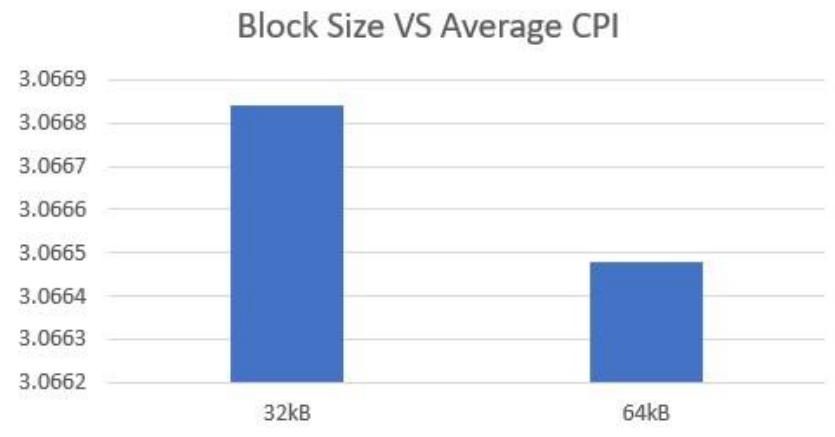
- By running 458.sjeng benchmark and the average CPI calculated, following conclusions can be made regarding the associativity:
- Both L1I and L1D caches the performance of the processor is directly affected by the associativity.
- **Conclusion:** As the associativity increases the CPI of the processor also decreases.

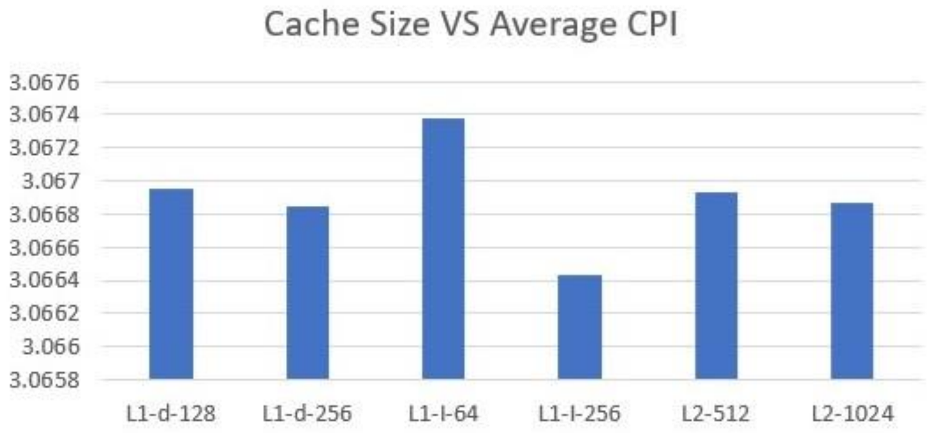
GRAPHICAL ANALYSIS OF L2 ASSOCIATIVITY FOR 458.SJENG



- By running the 458.sjeng benchmark the following observations about the L2 associativity were made:
- L2 cache when directly mapped has the lowest performance.
- L2 cache 8-way has the better performance then any other Associativity.
- **Conclusion:** Associativity of L2 cache is proportional to the performance.

GRAPHICAL ANALYSIS OF CACHE AND BLOCK SIZE FOR 458.SJENG





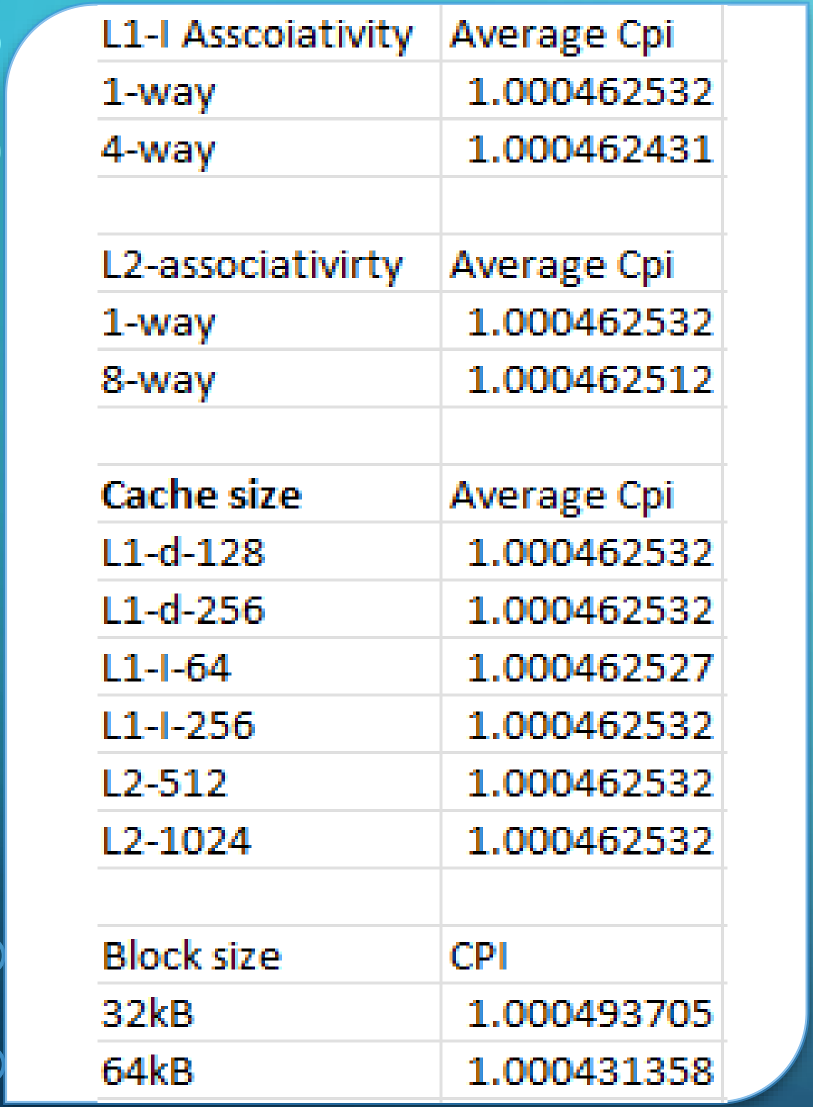
- 458.sjeng benchmark when run in system emulation mode, it is deduced that the block size has considerably more affect on the performance the than the cache size. We can see that CPI decreases when block size is doubled
- **Conclusion:** When it comes to size of multiple levels of Cache and the block, the performance is directly proportional to the Size of the Cache as well as the logical blocks.

OPTIMUM CPI FOR 470.LBM

- The minimum CPI obtained was : 1.000431353333333:
- The memory configuration is as follows:
  - Block size : 64
  - L1 Data Cache Associativity: 4
  - L1 Instruction Cache Associativity: 4
  - L2 Cache Associativity: 8
  - L1 Data Cache Size: 256kB
  - L1 Instruction Cache Size: 256kB
  - L2 Cache Size:1mB

ANOMALIES WITH 470.LBM

- The Graphical analysis of 470.lbm is not available because we found anomalies in the results obtained as shown in the figure.



L1-I Asscoiativity	Average Cpi
1-way	1.000462532
4-way	1.000462431
L2-associativirty	Average Cpi
1-way	1.000462532
8-way	1.000462512
Cache size	Average Cpi
L1-d-128	1.000462532
L1-d-256	1.000462532
L1-I-64	1.000462527
L1-I-256	1.000462532
L2-512	1.000462532
L2-1024	1.000462532
Block size	CPI
32kB	1.000493705
64kB	1.000431358

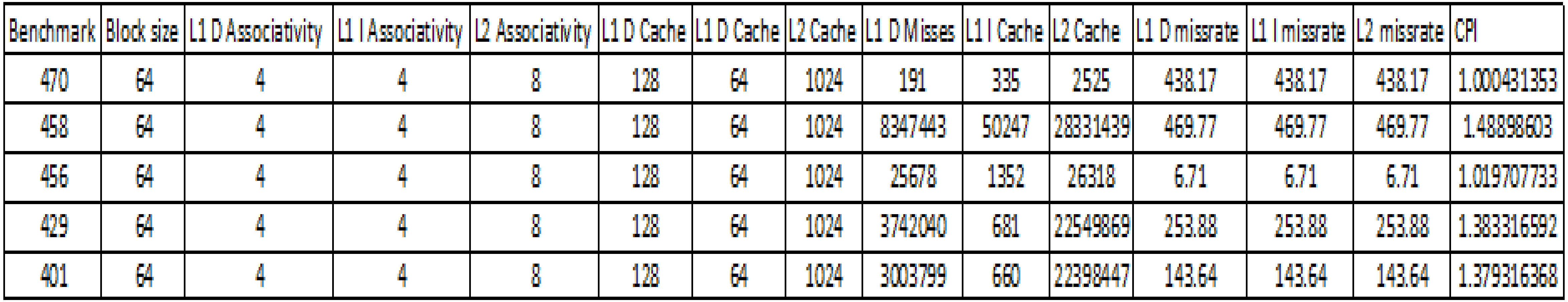
OPTIMUM CPI

- The minimum CPI is found for all the benchmarks, 401.bzip2, 429.mcf 456.hmmer, 458.sjeng, and 470.lbm, with the configuration which we considered is given below.
  - Block size : 64
  - L1 Data Cache Associativity: 4
  - L1 Instruction Cache Associativity: 4
  - L2 Cache Associativity: 8
  - L1 Data Cache Size: 256kB
  - L1 Instruction Cache Size: 256kB
  - L2 Cache Size: 1mB
- There are other configurations which have the same CPI value in respective benchmarks, but they were not considered as they are not consistent as the above configuration which had the minimum CPI in all the benchmarks

OPTIMUM CPI WITH CONDITIONS

- There were 2 given conditions, not having more than 256kB size for L1 and 1mB for L2. so, we considered L1 Data cache: 128kB and L1 Instruction cache: 64kB and L2 cache: 1mB
- With the prescribed conditions, we found the minimum CPI when simulated with the following configuration
  - Block size : 64
  - L1 Data Cache Associativity: 4
  - L1 Instruction Cache Associativity: 4
  - L2 Cache Associativity: 8
  - L1 Data Cache Size: 128kB
  - L1 Instruction Cache Size: 64kB
  - L2 Cache Size: 1mB

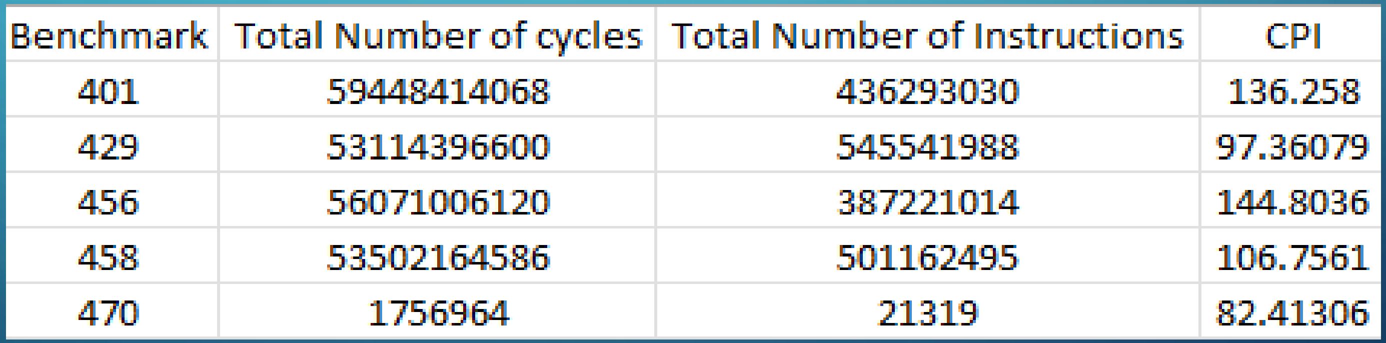
ACCUMULATED CONFIGURATIONS FOR THE OPTIMUM CPI WITH SEPARATED L1(256KB) AND UNIFIED L2(1MB)



Benchmark	Block size	L1 D Associativity	L1 I Associativity	L2 Associativity	L1 D Cache	L1 D Cache	L2 Cache	L1 D Misses	L1 I Cache	L2 Cache	L1 D missrate	L1 I missrate	L2 missrate	CPI
470	64	4	4	8	128	64	1024	191	335	2525	438.17	438.17	438.17	1.000431353
458	64	4	4	8	128	64	1024	8347443	50247	28331439	469.77	469.77	469.77	1.48898603
456	64	4	4	8	128	64	1024	25678	1352	26318	6.71	6.71	6.71	1.019707733
429	64	4	4	8	128	64	1024	3742040	681	22549869	253.88	253.88	253.88	1.383316592
401	64	4	4	8	128	64	1024	3003799	660	22398447	143.64	143.64	143.64	1.379316368

CPI VALUES CALCULATED WITH NO CACHE SPECIFIED WHILE BUILDING THE CPU

- We have calculated the CPI by dividing the number of cycles and the number of instructions to find the below values, however the values were found to be anomalous and hence no analysis was presented.



Benchmark	Total Number of cycles	Total Number of Instructions	CPI
401	59448414068	436293030	136.258
429	53114396600	545541988	97.36079
456	56071006120	387221014	144.8036
458	53502164586	501162495	106.7561
470	1756964	21319	82.41306

BETTER CACHE?

- If we need better performance, we consider larger sizes of the memory and higher Associativity in our memory hierarchy.
- But we should also consider that cost increases as the cache size increases.
- Benefits of larger cache is better hit rate and more storage for processing.
- Higher Associativity reduces the cache misses and conflict, but also increases the cache cost because of the hardware overhead.

COST FUNCTION

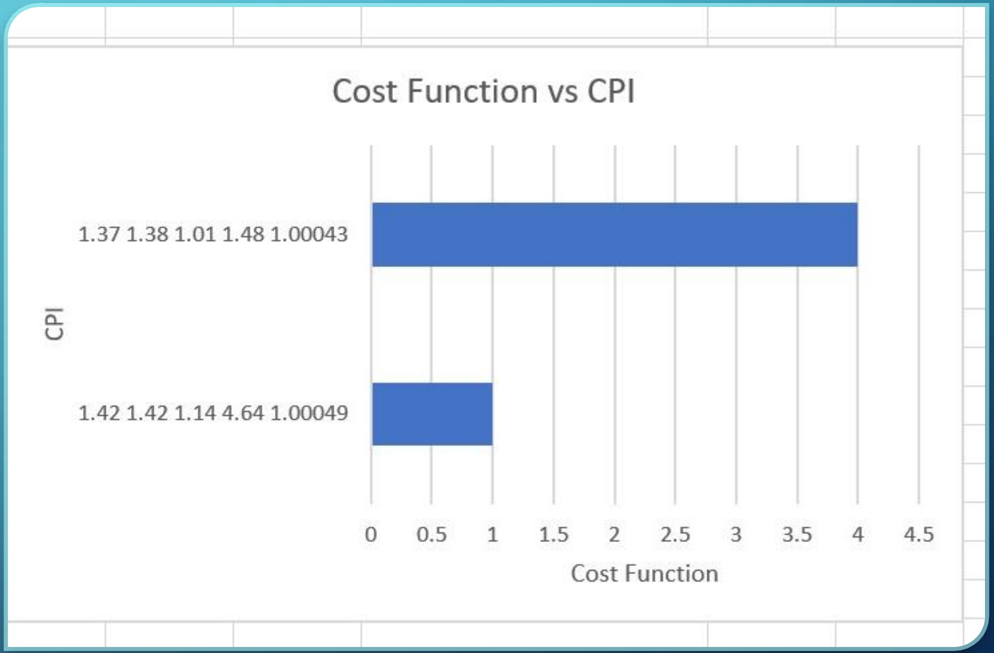
- Cost function is directly proportional to the size of the cache, hence larger the cache more is the cost. L1 cache is costlier per unit than L2 cache.
- So, we defined the cost function in the following way
- Cost function = (L1 Data cache size x L1 cache cost per unit)+(L1 Instruction cache size x L1 cache cost per unit)+(L2 cache size x L2 cache cost per unit)+(L1 Data cache Associativity x L1 Data cache Associativity cost)+(L1 Instruction cache Associativity x L1 Data cache Associativity cost)+(L2 cache Associativity x L2 cache Associativity cost)

OPTIMIZING THE COST

- According to an article by University of Texas at Austin, we found that the costs are as below in 2010:
  - Cost of L1 cache = 0.003$/kB
  - Cost of L2 cache = 0.00002$/kB
  - Cost of L1 Associativity = 0.25$/kB
  - Cost of L1 Associativity = 0.007$/kB
- For our optimum configuration mentioned in the page number:11, the cost is: 3.6125$
- We also found that the cost is in the range of 1-4$

GRAPHICAL REPRESENTATION OF COST VS CPI

- Configuration:
  - Block size : 64
  - L1 Data Cache Associativity: 4
  - L1 Instruction Cache Associativity: 4
  - L2 Cache Associativity: 8
  - L1 Data Cache Size: 256kB
  - L1 Instruction Cache Size: 256kB
  - L2 Cache Size: 1mB
- Cost = (256x 0.003)+(256 0.003)+(1024 x 0.00002)+(4 x 0.25)+(4x 0.25)+(8 x 0.007)
- Cost = 3.6125$/configuration



CONCLUSION

- When CPI is minimum the performance is better.
- CPI is minimum when the cache size is larger.
- CPI is inversely proportional to the Associativity.
- We observed that as CPI decrease, Cost function increases.
- Which means performance comes with the cost.